CSCI 3060U/SOFE3980U – Phase One

Software Quality Assurance

James Gelinas	100486790
Henry Julian	100502703
Matthew Clark	100429906

February 11th 2015

We, certify that this work is our own, submitted for CSCI 4210U in compliance with the UOIT Academic Integrity Policy.

Part One

Transaction	Test Name	Intention
Login	login0	check if a current user is logged in
	login1	Asks user for username
	login2	Check if username exists
	login3	reads in the users account file
	login4	Check if password matches the username password
	login5	Test if we can login as a normal user
	login6	Test if we can log in as an admin
	login7	Test that we cannot make privileged transactions as a standard user
	login8	Test that we can make privileged transactions as an admin user
Logout	logout0	check if a user is currently logged in.
	logout1	Write out the daily transaction file
	logout2	Test if we can make another transaction
Create	create0	check if a user is logged in
	create1	Ask user for username (Max 15 characters)
	create2	Check if username is unique
	create3	Ask user for a password (min 8 character, one must be uppercase)
	create4	Ask user type(Full, Buy, Sell)

	create5	verify information has been saved to the daily transaction file
	create6	Verify that there is a limit of $999,999 that can be credited to the account.
Delete	delete0	Check that no further transactions can be conducted by deleted user
	delete1	Ask user for user account to be deleted
	delete2	Verify that the user account to be deleted exists
	delete3	Remove tickets for sale and pending purchases associated with the user account
	delete4	Delete the user account and check that it no longer exists
Sell	sell0	Verify that the user has proper privileges
	sell1	Verify that the event title given is valid (Max 25 characters)
	sell2	Verify that the given sales price is valid (Max 999.99)
	sell3	Check that number of tickets for sale is valid (Max 100)
	sell4	Check that all information is stored into the daily transaction file
	sell5	Check that no other transactions can occur until a new sell session is started
Buy	buy0	Verify that the user has proper privileges
	buy1	Verify that no more than four tickets can be buy each transaction
	buy2	Check the current title and enough ticket available
	buy3	Verify the confirmation form of yes or no for purchasing the ticket

Refund	refund0	Check for buyer’s and seller’s username and the amount of credit to transfer
	refund1	Verify both buyer and seller as current users
	refund2	Check if the information saved for the daily transaction file
	refund3	Check the transfer amount of credit from seller’s credit balance to the buyer’s credit balance
Addcredit	addcredit0	Check that the user is in admin mode
	addcredit1	Verify the username and the amount of credit is added to that username.
	addcredit2	Check if the information saved in daily transaction file
	addcredit3	Verify that maximum $1000.00 can be added to an account

Test Organization

The tests will be organized into directories based on which system they are testing, the file tree follows below. The tests will be saved in files which correspond with the transaction name, login0, login1, login2, logout0, buy1, etc. Input files will be saved in their corresponding test folders will the extension “.inp”. Depending on if a transaction will get created or if there is terminal output, or both, these will be saved in their respective transaction name with the file type “.etf” for transactions, “.bto” for terminal output and finally the actual output for the transaction file and terminal output will be saved in a “.aetf” for transactions and “.abto” for terminal output.

```
Tests
        -Login
        -Logout
        -Create
        -Delete
        -Sell
        -Buy
        -Refund
```

-Addcredit

Test Run Plan

Test will be run using a BASH script which will run the front end application. For each test input file the script will compare it with expected output for each test input. If there is a difference between the expected output and the actual output of the system there will be a warning created to analyse the difference between the expected output and the generated output.

Testing will occur as the software is developed. Each different function of the front end will be created one at a time and will be tested as a single class or module before it is incorporated into the complete project. Each test will have two sub tests, one in where the expected output will be a successful transaction and one where we expect an unsuccessful transaction or an error to occur. These are both essential in order to cover all of the bases as not everything always runs 100 percent perfectly every time. One the entire front end is put together we will once again run all of the tests in order to ensure that everything works in unison.

Requirements Problems

Problem 1: There is nothing in the requirements about automatically logging an account out after a certain amount of inactivity or after navigating through the site. In order to deal with this ambiguity we have decided to put a timer of 15 minutes on each login session, after this timer has run out the account will be automatically logged out.

Problem 2: There is no requirement preventing an account from being logged in to more than one session, in order to handle that occurrence the account will be logged off of any other sessions before they are logged in to the new session.

Problem 3: When creating a new user account there is no limitations on what the passwords can be, therefore we cannot anticipate how many characters we need storage for. In order to account for this a specification will be put when creating an account that a password can be at most 12 characters with letters and numbers only.

Problem 4: In the buy section of the requirements that specify what should happen if there are insufficient funds in the buyers account, to handle this a check will be run before the transaction is processed and an error will be reported if there is not enough money in the users account to cover the price of the tickets.